

Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1-25 (previously cancelled)

26. (currently amended) A system for distributed file storage comprising:

a plurality of servers providing, to a plurality of clients, file access services for accessing files stored on the plurality of servers; and

a list of neighbor servers maintained by each server, wherein the neighbor servers are grouped as a subset of the plurality of servers,

wherein each file is stored in the form of a plurality of N pieces ~~on N servers of equal size~~, the N pieces being generated from the file ~~and stored on the plurality of servers~~, and

wherein the list is used to obtain information for reconstructing files stored on the neighbor servers, such that any K out of the N pieces can be used to reconstruct any file,

wherein a server belonging to more than one group acts as a boundary server, and wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected.

27. (previously presented) The system of claim 26, wherein the servers use a peer-to-peer network for communication with each other.

28. (previously presented) The system of claim 26, wherein the files are stored using a peer-to-peer network.

29. (canceled)

30. (previously presented) The system of claim 26, wherein at least one of the servers is a logical server that includes a plurality of physical servers connected via a network.

31. (previously presented) The system of claim 26, wherein each server belongs to a group defined by its corresponding list,

wherein a server that belongs to multiple groups maintains corresponding multiple lists for each group to which it belongs, and

wherein, upon receiving a request for a file, the server distributes the request to each of the multiple groups.

32. (previously presented) The system of claim 26, wherein none of the pieces is unique.

33. (currently amended) A method for distributed file storage comprising:

dividing a plurality of servers into a plurality of groups, with each server belonging to at least one group;

on each server, maintaining a list of neighbor servers belonging to the same group,
wherein each group is a subset of the plurality of servers;

supporting file access services on each of the servers;

transforming a file into a plurality of N pieces ~~of equal size~~ that are derived from the file,
such that any K out of the N pieces can be used to reconstruct any file; and

storing each of the pieces on N servers selected from the list,

wherein a server belonging to more than one group acts as a boundary server, and

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected.

34. (previously presented) The method of claim 33, further comprising using a peer-to-peer network for communication between the servers.

35. (previously presented) The method of claim 33, further comprising using a peer-to-peer network for storing the file.

36. (previously presented) The method of claim 33, further comprising maintaining functional equivalence among the servers.

37. (previously presented) The method of claim 33, further comprising verifying availability of the neighbor servers.

38. (previously presented) The method of claim 33, wherein the list is a dynamic list, and further comprising:

polling the servers on the dynamic list of the neighbor servers and measuring a network distance for each server;

keeping each available server on the dynamic list in the same group;

removing unavailable servers from the dynamic list of the same group; and

switching at least one server of the plurality of servers into a neighbor group of servers if the measured network distance is lower than a predetermined threshold.

39. (previously presented) The method of claim 38, wherein better network connectivity is determined based on the network distance.

40. (previously presented) The method of claim 38, further comprising:

polling the dynamic list;

connecting a client to a server on the dynamic list;

switching the client to a different server on the dynamic list that has a smaller workload or lower network distance relative to the client; and

delivering to the client a file requested by the client that is stored on the servers.

41. (previously presented) The method of claim 38, further comprising:

polling the dynamic list;

connecting a client to a server on the dynamic list;

switching the client to a different server on the dynamic list that has a lower network distance from a perspective of the client; and

delivering to the client a file requested by the client that is stored on the servers.

42. (previously presented) The method of claim 33, wherein the list is a static list of servers.

43. (previously presented) The method of claim 33, further comprising:

maintaining multiple lists for a server that belongs to multiple groups; and,

upon receiving a request for a file, distributing the request to servers selected from the multiple groups.

44. (canceled)

45. (currently amended) A method of accessing files in a distributed file storage system comprising:

dividing a plurality of the servers into a plurality of groups, wherein each server belongs to at least one group, and wherein each group is a subset of the plurality of servers;

supporting file access services on each of the servers for accessing a file stored on the servers;

at each server, maintaining a list of neighbor servers that belong to the same group;

generating a plurality of N pieces ~~of equal size~~ from the file; and

distributing the plurality of N pieces to the neighbor servers in the same group in order to achieve a desired fault tolerance level, wherein the fault tolerance level is defined by how many servers (K) out of the total number of N servers on which the N file pieces are stored, can fail, such that any K out of the N pieces can be used to reconstruct any file,

wherein a server belonging to more than one group acts as a boundary server, and

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected.

46. (previously presented) The method of claim 45, further comprising using a peer-to-peer network for communication between the servers.

47. (canceled)

48. (previously presented) The method of claim 45, further comprising maintaining functional equivalence of the servers for purposes of storing the file.

49. (previously presented) The method of claim 45, further comprising verifying availability of the neighbor servers.

50. (previously presented) The method of claim 45, further comprising:

polling the list of neighboring group servers and measuring a relative network distance for each client-server pair;

connecting a client to any of the servers on the list using the measured metric;
switching the client to a server on the list that has a lower relative network distance and a smaller workload to improve a level of service; and
delivering, to the client, the file requested by the client.

51. (previously canceled)

52. (previously presented) The method of claim 45, further comprising:

polling servers on the list that belong to the same group;
connecting a client to any of the servers on the list that belong to the same group;
switching the client to a server on the list that has a smaller workload; and
delivering, to the client, the file that is requested by the client.

53. (previously presented) The method of claim 52, further comprising:

identifying a name of the file in namespace;
sending a request for the file from the client to the server to which the client is connected;
sending a request for the file from the server to which the client is connected to a neighbor server;
distributing the request to other servers if the server to which the client is connected cannot provide the file pieces information;
checking for file pieces in a local cache of each server that received the request;
sending the pieces to the server to which the client is connected;
transferring the pieces to the client; and
assembling the file on the client.

54. (previously presented) The method of claim 52, further comprising sending all the pieces from the neighbor servers to the server to which the client is connected simultaneously.

55. (previously presented) The method of claim 45, wherein the list is a static list of servers.

56. (canceled) The method of claim 45, wherein none of the pieces is unique.

57. (currently amended) A method of naming files in a distributed file storage system comprising:

dividing a plurality of servers into a plurality of groups such that each server belongs to at least one group, wherein each group is a subset of the plurality of servers;

supporting file access services on each of the servers for accessing files stored on the servers;

giving file names for the files uniformly and independent of location of the files on the servers;

storing the files on N of the servers using the names, wherein each file is transformed into a plurality of pieces ~~of equal size~~ that are generated from the file, such that any K out of the N pieces can be used to reconstruct any file,

wherein a server belonging to more than one group acts as a boundary server, and

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected; and

accessing the files using the file access services from any of servers.

58. (previously presented) The method of claim 57, further comprising using a peer-to-peer network for communication between the servers

59. (previously presented) The method of claim 57, further comprising maintaining functional equivalence of the servers.

60. (previously presented) The method of claim 57, further comprising verifying availability of the neighbor servers.

61. (previously presented) The method of claim 57, further comprising:
on each server, maintaining a dynamic list of the neighbor servers in the same group;
polling servers on the dynamic list;
connecting a client to any of the servers on the dynamic list;
switching the client to a server on the dynamic list that has lower network distance to improve level of service; and

delivering to the client a file requested by the client that is stored on the servers.

62. (canceled)

63. (previously presented) The method of claim 57, further comprising:
on each server, maintaining a dynamic list of the neighbor servers;
polling servers on the dynamic list that belong to the same group;
connecting a client to any of the servers on the dynamic list;
switching the client to a server on the dynamic list that has a smaller workload; and
delivering to the client a file requested by the client that is stored on the servers.

64. (previously presented) The method of claim 57, further comprising:

maintaining a uniform name space as a tree with a common root and a logical path to each stored file as part of the distributed file storage system; and

maintaining data files and directory files to be stored in the distributed file storage system.

65. (previously presented) The method of claim 64, further comprising:

maintaining the directory files as executable files with their own executable code and data; and

providing translation from a logical path inside the uniform name space to a unique file identifier using the directory files.

66. (currently amended) A system for organizing distributed file storage comprising:

N functionally equivalent servers each providing file access services, for a plurality of clients, to files stored on the servers, such that when a file is divided into N pieces stored on the N servers, any K out of the N servers can be used to reconstruct the file; and

each file being transformed into the N pieces that are generated from the file and stored on the N servers,

wherein information for reconstructing the files is obtained from the [[N]] servers,

wherein the servers are organized into groups, each group being a subset of the plurality of N servers, and wherein a server belonging to more than one group acts as a boundary server,
and

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected.

67. (previously presented) The system of claim 66, further comprising a dynamic list of neighbor servers maintained by each server,

wherein the neighbor servers are a subset of the plurality of servers, and

wherein the dynamic list is used to obtain the information for reconstructing the files.

68. (previously presented) The system of claim 66, further comprising a static list of neighbor servers maintained by each server,

wherein the neighbor servers are a subset of the plurality of servers, and

wherein the static list is used to obtain the information for reconstructing the files.

69. (currently amended) A computer program product for distributed file storage, the computer program product comprising a computer useable medium having computer program logic recorded thereon for controlling a processor, the computer program logic comprising:

computer program code means for ~~dividing~~ organizing a plurality of servers into a plurality of groups, with each server belonging to at least one group;

on each server, computer program code means for maintaining a list of neighbor servers belonging to the same group;

computer program code means for supporting file access services on each of the servers;

computer program code means for transforming a file into a plurality of pieces that are derived from the file, such that any K out of the N pieces can be used to reconstruct any file; and

computer program code means for storing each of the pieces on N servers selected from the list,

wherein a server belonging to more than one group acts as a boundary server, and

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected.

70. (currently amended) A method of accessing files in a distributed file storage system comprising:

~~dividing~~ organizing a plurality of the servers into dynamically reconfigurable groups, wherein each server belongs to at least one group that is reconfigurable based on minimum network distance, wherein each group is a subset of the plurality of servers;

supporting file access services on each of the servers for accessing a file distributed among the servers;

generating a plurality of N pieces ~~of equal size~~ from the file; and

distributing the plurality of N pieces to N of the neighbor servers in the same group in order to achieve a desired fault tolerance level based on how many servers (K) out of the plurality of servers are available, such that any K out of the N pieces can be used to reconstruct any file.

wherein a server belonging to more than one group acts as a boundary server, and

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected.

71. (new) The system of claim 26, wherein the servers are organized into a plurality of groups such that server-to-server response time between any two servers of the same group does not exceed a predetermined limit, and

wherein, when a new server is connected to one of the groups, the new server uses information of the boundary servers to connect to a group so as to have optimal server-to-server response time to its neighbors within its group.

72. (new) The system of claim 71, wherein, when one server is disconnected from the system, the remaining servers use information of the boundary servers to reconfigure their groups so as to have optimal server-to-server response times to their neighbors within their groups.

73. (new) The method of claim 33, wherein, when one server is disconnected from the system, the remaining servers use information of the boundary servers to reconfigure their groups so as to have optimal server-to-server response times to their neighbors within their groups.

74. (new) A system for distributed file storage comprising:
a plurality of servers providing, to a plurality of clients, file access services for accessing files stored on the plurality of servers;

wherein the servers are organized into a plurality of groups such that server-to-server response time between any two servers of the same group does not exceed a predetermined limit;

wherein a server belonging to more than one group acts as a boundary server;

wherein, when a new server is connected to one of the groups, the new server uses information of the boundary servers to connect to a group so as to have optimal server-to-server response time to its neighbors within its group;

wherein, when one server is disconnected from the system, the remaining servers use information of the boundary servers to reconfigure their groups so as to have optimal server-to-server response times to their neighbors within their groups; and

a list of neighbor servers of each server's group maintained by each server, wherein the neighbor servers are a subset of the plurality of servers,

wherein each file is stored in the form of a plurality of N pieces, the pieces being generated from the file and stored on N servers of the plurality of servers,

wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected,

wherein the list is used to obtain information for reconstructing files stored on the neighbor servers, and

wherein any client can use any of the plurality of servers to access any of its files stored on the system.